T'D

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/708,722	11/09/2000	Stephan J. Jourdan	2207/9800	2194

25693 7590 06/16/2006

KENYON & KENYON LLP
RIVERPARK TOWERS, SUITE 600
333 W. SAN CARLOS ST.
SAN JOSE, CA 95110

EXAMINER	
LI, AIMEE J	

ART UNIT	PAPER NUMBER
2183	

DATE MAILED: 06/16/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	**Application No.** 09/708,722	**Applicant(s)** JOURDAN ET AL.
	Examiner Aimee J. Li	**Art Unit** 2183

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *03 April 2006*.

2a)☒ This action is **FINAL**. 2b)□ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-19* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-19* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) □ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application (PTO-152)

6) □ Other: _____.

DETAILED ACTION

1. Claims 1-19 have been examined.

Papers Submitted

2. It is hereby acknowledged that the following papers have been received and placed on record in the file: Appeal Brief as filed 07 September 2005.

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claims 1, 3-8, 12-15 and 17-19 are rejected under 35 U.S.C. 103(a) as being unpatentable over Patel et al., *Improving Trace Cache Effectiveness with Branch Promotion and Trace Packing*, in further view of Johnson, U.S. Patent No. 5,924,092.

5. Regarding claim 1, Patel has taught a cache comprising a cache line to store an instruction segment further comprising a plurality of instructions stored in sequential positions of the cache line (see Col.1 line 26 – Col.2 line 15). Patel has not explicitly taught storing the plurality of instructions in sequential positions of a cache line in reverse program order.

6. However, Johnson has taught the storing of blocks of data in reverse order so that those blocks that were in the first block of a data structure that are frequently accessed and modified will require less moving and fewer modifications after being placed at the last location of data structure, the fewer modifications resulting in improved performance (see Johnson, Col.4 lines 13-24). Because the traces of Patel are indexed, as well as accessed and modified, via their head

(first) instructions (see Patel, Col.4 lines 23-31 and Col.5 lines 12-18), one of ordinary skill in the art would have found it obvious to modify the instruction segment of Patel to store the instructions of the instruction trace in reverse order so that the frequently accessed and modified head of the trace will be moved and modified fewer times so that performance is improved.

7. Regarding claim 3, Patel in view of Johnson has taught the cache of claim 1, wherein the instruction segment is a trace (see Patel, Col.3 lines 2-12).

8. Regarding claim 4, Patel in view of Johnson has taught the cache of claim 1, wherein the instruction segment is a basic block (see Patel, Col.2 lines 2-5).

9. Regarding claim 5, Patel has taught a segment cache (see "trace cache" of Fig.1) for a front-end system in a processor, comprising a plurality of cache entries to store instruction segments (see Col.1 line 26 – Col.2 line 15). Patel has not explicitly taught storing the instruction segments in reverse program order.

10. However, Johnson has taught the storing of blocks of data in reverse order so that those blocks that were in the first block of a data structure that are frequently accessed and modified will require less moving and fewer modifications after being placed at the last location of data structure, the fewer modifications resulting in improved performance (see Johnson, Col.4 lines 13-24). Because the traces of Patel are indexed, as well as accessed and modified, via their head (first) instructions (see Patel, Col.4 lines 23-31 and Col.5 lines 12-18), one of ordinary skill in the art would have found it obvious to modify the instruction segment of Patel to store the instructions of the instruction trace in reverse order so that the frequently accessed and modified head of the trace will be moved and modified fewer times so that performance is improved.

11. Regarding claim 6, Patel in view of Johnson has taught an apparatus comprising:

a. An instruction cache system (see Patel, "instruction cache" of Fig.1),

b. An instruction segment system, comprising:

i. A fill unit (see Patel, "fill unit" of Fig.1) provided in communication with the instruction cache system, the segment cache of claim 5 included therein (see Patel, Fig.1),

c. A selector (see Patel, "selection logic" of Fig.1) coupled to an output of the instruction cache system and to an output of the segment cache (see Patel, Fig.1).

12. Regarding claim 7, Patel in view of Johnson has taught the front-end system of claim 6, wherein the instruction segment system further comprises a segment predictor (see Patel, "multiple branch predictor" of Fig.1) provided in communication with the segment cache. Here, when the multiple branch predictor is coupled with the trace cache and mediated by the selection logic, it effectively predicts segments because the basic blocks stored in the trace cache all begin with branches (see Patel, Col.5 lines 5-29).

13. Regarding claim 8, Patel has taught a method for storing instruction segments in a processor, comprising:

a. Building an instruction segment based on program flow (see Col.1 line 26 – Col.2 line 15),

b. Storing the instruction segment in a cache (see Col.1 line 26 – Col.2 line 15).

14. Patel has not explicitly taught wherein the instruction segment is stored in reverse program order.

15. However, Johnson has taught the storing of blocks of data in reverse order so that those blocks that were in the first block of a data structure that are frequently accessed and modified

will require less moving and fewer modifications after being placed at the last location of data structure, the fewer modifications resulting in improved performance (see Johnson, Col.4 lines 13-24). Because the traces of Patel are indexed, as well as accessed and modified, via their head (first) instructions (see Patel, Col.4 lines 23-31 and Col.5 lines 12-18), one of ordinary skill in the art would have found it obvious to modify the instruction segment of Patel to store the instructions of the instruction trace in reverse order so that the frequently accessed and modified head of the trace will be moved and modified fewer times so that performance is improved.

16. Regarding claim 12, Patel in view of Johnson has taught the method of claim 8, wherein the instruction segment is a trace (see Patel, Col.3 lines 2-12).

17. Regarding claim 13, Patel in view of Johnson has taught the method of claim 8, wherein the instruction segment is a basic block (see Patel, Col.2 lines 2-5).

18. Regarding claim 14, Patel has taught a processing engine, comprising:

a. A front-end stage to build and store instruction segments (see Col.1 line 26 – Col.2 line 15),

b. An execution unit (see "HPS Execution Core" in Fig.1) in communication with the front end stage (see Col.5 line 30 – Col.6 line 4).

19. Patel has not explicitly taught building and storing the instruction segments in reverse program order.

20. However, Johnson has taught the storing of blocks of data in reverse order so that those blocks that were in the first block of a data structure that are frequently accessed and modified will require less moving and fewer modifications after being placed at the last location of data structure, the fewer modifications resulting in improved performance (see Johnson, Col.4 lines

13-24). Because the traces of Patel are indexed, as well as accessed and modified, via their head (first) instructions (see Patel, Col.4 lines 23-31 and Col.5 lines 12-18), one of ordinary skill in the art would have found it obvious to modify the instruction segment of Patel to store the instructions of the instruction trace in reverse order so that the frequently accessed and modified head of the trace will be moved and modified fewer times so that performance is improved.

21. Regarding claim 15, Patel in view of Johnson has taught the processing engine of claim 14, wherein the front-end stage comprises:

a. An instruction cache system (see Patel, “instruction cache” of Fig.1),

b. An instruction segment system, comprising:

i. A fill unit (see Patel, “fill unit” of Fig.1) provided in communication with the instruction cache system (see Patel, Fig.1),

ii. A segment cache (see “trace cache” of Fig.1),

c. A selector (see Patel, “selection logic” of Fig.1) coupled to an output of the instruction cache system and to an output of the segment cache (see Patel, Fig.1).

22. Regarding claim 17, Patel in view of Johnson has taught the method of claim 15, wherein the instruction segments are traces (see Patel, Col.3 lines 2-12).

23. Regarding claim 18, Patel in view of Johnson has taught the method of claim 15, wherein the instruction segments are basic blocks (see Patel, Col.2 lines 2-5).

24. Regarding claim 19, Patel in view of Johnson has taught the method of claim 15, wherein the instruction segment cache system further comprises a segment predictor (see Patel, “multiple branch predictor of Fig.1) provided in communication with the segment cache. Here, when the multiple branch predictor is coupled with the trace cache and mediated by the selection logic, it

effectively predicts segments because the basic blocks stored in the trace cache all begin with branches (see Patel, Col.5 lines 5-29).

25. Claims 2, 9-11 and 16 are rejected under 35 U.S.C. 103(a) as being unpatentable over Patel et al., *Improving Trace Cache Effectiveness with Branch Promotion and Trace Packing*, in view of Johnson, U.S. Patent No. 5,924,092, in further view of Peled et al., U.S. Patent No. 6,076,144.

26. Regarding claim 2, Patel in view of Johnson has taught the cache of claim 1, but has not explicitly taught wherein the instruction segment is an extended block.

27. However, Peled has taught trace segments which have multiple entry points and a single exit that allow redundant code segments to be eliminated from the trace cache, thereby improving cache utilization (see Peled, Col.1 lines 60-63, Col.4 lines 13-37, and Fig.3). Because the specification has defined an extended block to have multiple entry points and a single exit point (see p.2 of Specification), one of ordinary skill in the art would have found it obvious to modify the instruction segments of Patel to allow for multiple entry points and a single exit so that redundant code segments could be eliminated from the trace cache and performance could be improved.

28. Regarding claim 9, Patel in view of Johnson have taught the method of claim 9, but have not explicitly taught wherein the method further comprises:

a. Building a second instruction segment based on program flow,

b. If the first and second instruction segments overlap, extending the first instruction segment to include non-overlapping instructions from the second instruction segment.

29. However, Peled has taught building a second instruction segment based on program flow and subsequently extending the first instruction segment to include the non-overlapping instructions from the second instruction segment if the two segments overlap (see Col.4 lines 13-37) in order to reduce the degree of code redundancy in the trace cache (see Col.1 lines 60-63). One of ordinary skill in the art would have recognized that it is desirable to reduce redundancy within a trace cache so that the cache can be more effectively used and more different traces stored. Therefore, one of ordinary skill in the art would have found it obvious to extend an existing instruction segment to include non-overlapping instructions from a second instruction segment in order to reduce trace cache redundancy.

30. Regarding claim 10, Patel in view of Johnson in further view of Peled has taught the method of claim 9, but has not explicitly taught wherein the extending comprises storing the non-overlapping instructions in the cache in reverse program order in successive cache positions adjacent to the instructions from the first instruction segment.

31. However, Patel in view of Johnson has taught that instructions in instruction segments are stored in reverse program order (see paragraphs 21-23 above). Because, an extended segment is still an instruction segment, one of ordinary skill in the art would have found it obvious to also store the extended instruction segments in reverse program order.

32. Regarding claim 11, Patel in view of Johnson has taught the method of claim 8, but has not explicitly taught wherein the instruction segment is an extended block.

33. However, Peled has taught trace segments which have multiple entry points and a single exit that allow redundant code segments to be eliminated from the trace cache, thereby improving cache utilization (see Peled, Col.1 lines 60-63, Col.4 lines 13-37, and Fig.3). Because

the specification has defined an extended block to have multiple entry points and a single exit point (see p.2 of Specification), one of ordinary skill in the art would have found it obvious to modify the instruction segments of Patel to allow for multiple entry points and a single exit so that redundant code segments could be eliminated from the trace cache and performance could be improved.

34. Regarding claim 16, Patel in view of Johnson has taught the method of claim 15, but has not explicitly taught wherein the instruction segments are extended blocks.

35. However, Peled has taught trace segments which have multiple entry points and a single exit that allow redundant code segments to be eliminated from the trace cache, thereby improving cache utilization (see Peled, Col.1 lines 60-63, Col.4 lines 13-37, and Fig.3). Because the specification has defined an extended block to have multiple entry points and a single exit point (see p.2 of Specification), one of ordinary skill in the art would have found it obvious to modify the instruction segments of Patel to allow for multiple entry points and a single exit so that redundant code segments could be eliminated from the trace cache and performance could be improved.

Response to Arguments

36. Examiner withdraws rejections under 35 U.S.C. 101 in favor of the amended claims.

37. Applicant's arguments filed 03 April 2006 have been fully considered but they are not persuasive.

38. Applicant argues in essence on pages 5 and 9-10 "...there must be some suggestion or motivation to modify the reference or combine the reference teachings." This has not been persuasive. As is in the rejection above, the motivation to combine was found in Johnson,

specifically Johnson column 4, lines 14-24. Johnson specifically states "...In general, this results in a lower average number of updated frames per data modification, and thus improved overall performance."

39. Applicant argues in essence on page 6

> ...However, Patel neither teaches nor suggests that the fetch rates may be improved by reversing the order of the instructions in the traces. Applicants submit there would be no motivation to do so, since reversing instruction order would not appear to improve fetch rates given the teaching of Patel.

40. This has not been found persuasive. Johnson was relied upon to teach reversing instruction order, not Patel, so the motivation to combine would be found in Johnson, as is shown in the rejection above and response to the argument on page 5 above. Arguing that Patel has not taught the motivation would be contrary to the purpose of combining, since, if Patel did contain a motivation for reversing instruction order, Patel would have taught reversing instruction order and the second reference of Johnson would not have been needed. However, if Johnson had not provided motivation, In response to applicant's argument that there is no suggestion to combine the references, the examiner recognizes that obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. See *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988)and *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992).

41. Applicant argues in essence on pages 6-7

> ...such access of the trace would hinder the completion, thereby defeating the purpose of the trace.

42. This has not been found persuasive. There is nowhere stated in Patel that the trace cache will not function with its data in reverse program order not is there a statement in Johnson stating that reverse program order would render trace caches inoperable. In fact, Johnson states explicitly in column 4, lines 27-29 "...it should be appreciated that the elements may represent practically any types of memory blocks or segments, having any fixed or variable size." Therefore, this argument is mere allegation unless Applicant can identify a specific location within Patel and/or Johnson stating that a trace cache will not operate correctly when its elements are in reverse program order. Also, it seems that Applicant's are literally importing Johnson's invention into Patel's invention. However, the purpose of a combination is what it would have suggested to a person of ordinary skill in the art, not whether literally importing one invention to another would operate correctly. In response to applicant's argument that the modification [reverse program order] would result in a different operation, the test for obviousness is not whether the features of a secondary reference may be bodily incorporated into the structure of the primary reference; nor is it that the claimed invention must be expressly suggested in any one or all of the references. Rather, the test is what the combined teachings of the references would have suggested to those of ordinary skill in the art. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981).

43. Applicant's argue in essence on pages 7-8 and 9

> As such, a person of ordinary skill in the art would not be motivated to reverse the instructions in a trace, while using trace packing, to improve fetch rates.

44. This has not been found persuasive. As stated in the above rejection and responses, the motivation to combine the reverse program order is found within Johnson, not Patel. Patel's motivation for creating his trace cache is not the motivation relied upon. The motivation for combining the reverse program order is laid out in Johnson column 4, lines 14-24, which explicitly states "...In general, this results in a lower average number of updated frames per data modification, and thus improved overall performance."

45. Applicant's argue in essence on pages 8-9

> The Johnson reference is not directed toward improving performance during data access or use at all, but rather limited to discussing improving performance during data modification.

46. This has not been found persuasive. "Modification" means to "the act of modifying (dictionary.com "modification" ©2000)" and "modify" means "to change in form or character; alter (dictionary.com "modify" ©2000)." Therefore, "data modification" means the act of changing or altering data. In order to modify data, the data must be read from the memory, so that the processor knows what the data is and altering the data would require using the data in some process. Consequently, to say that Johnson "is not directed toward improving performance during data access or use at all" and then state "...limited to discussing improving performance during data modification" is a contradiction. Also, the argument that the Johnson does not relate to data access and use to improve performance is not reflected within the claim language. There is nothing in the claim language about improving performance during data access or use. In response to applicant's argument that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., improving performance

during data access or use) are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

Conclusion

47. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

48. A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

49. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Aimee J. Li whose telephone number is (571) 272-4169. The examiner can normally be reached on M-T 7:00am-4:30pm.

50. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

51. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications

may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

AJL
Aimee J. Li
8 June 2006

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100